Performance Characterization of Space Communications and Navigation (SCaN) Network by Simulation

Esther Jennings, David Heckman
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109
Esther.Jennings@jpl.nasa.gov, David.P.Heckman@jpl.nasa.gov

Abstract—As future space exploration missions will involve larger number of spacecraft and more complex systems, theoretical analysis alone may have limitations on characterizing system performance and interactions among the systems. Simulation tools can be useful for system performance characterization through detailed modeling and simulation of the systems and its environment.

To simulate Space-based networks, we have developed the Multi-mission Advanced Communications Hybrid Environment for Test and Evaluation (MACHETE) tool which captures the unique characteristics of space networking for performance assessment and end-to-end test evaluation. This tool contains orbital and planetary motion kinematics models, link engineering models, traffic load generation models and communication protocol models.

This paper reports the simulation of the Orion (Crew Exploration Vehicle) to the International Space Station (ISS) mission where Orion is launched by Ares into orbit on a 14-day mission to rendezvous with the ISS. Communications services for the mission are provided by the Space Communication and Navigation (SCaN) network infrastructure which includes the NASA Space Network (SN), Ground Network (GN) and NASA Integrated Services Network (NISN). The objectives of the simulation are to determine whether SCaN can meet the communications needs of the mission, to demonstrate the benefit of using QoS prioritization, and to evaluate network key parameters of interest such as delay and throughput.[1][2]

TABLE OF CONTENTS

1. INTRODUCTION

In NASA's exploration missions to the Solar System, incorporating network technologies for space-based networking is predicted to greatly increase the efficiency of these missions. To support communications and navigation requirements of future missions, the Space Communications and Navigation (SCaN) Program Office was created within the Space Operations Mission Directorate (SOMD) to implement the needed end-to-end communications and tracking infrastructure. This infrastructure includes the NASA Space Network (SN), Ground Network (GN), and Deep Space Network (DSN) and NASA Integrated Services Network (NISN). Many new requirements for the SCaN network are currently driven by NASA's Constellation program which will take humans to the Moon and Mars. As future missions will involve greater numbers of spacecraft and systems, interactions and trade-offs among systems will be complex. As systems become more complex, theoretical analysis and direct experimentation may have limitations and may not capture the inter-dependencies or interactions among the systems. Simulation is the process of designing a model of a real system, conducting experiments with this model to understand the behavior of the system, and evaluating various strategies for the operation of the system. Simulation tools can add more details by incorporating environmental effects and interactions with other systems, and we can study a large number of system configurations that are complex, stochastic, and dynamic. In addition to simulation of communications, we are using numerical analysis (linear algebra) to compute real contact dynamics (geometry).

At Jet Propulsion Laboratory, we have developed the Multi-mission Advanced Communications Hybrid Environment for Test and Evaluation (MACHETE) simulation tool that captures the unique characteristics of space networking for performance assessment and end-to-end test evaluation. In this paper, we describe the use of MACHETE to model and simulate the Orion mission which is among Phase 1 missions of NASA's Constellation program. Orion mission will be the first manned mission of Orion (Crew

1
[1] 1-4244-1488-1/08/$25.00 ©2008 IEEE.
[2] IEEEAC paper#1132, Version 6, Updated 2008:01:19

Exploration Vehicle) that is launched by Ares (Crew Launch Vehicle) into a 51.6 degree inclined orbit on a 14 day mission to rendezvous with the International Space Station (ISS). The objective of the simulation is to evaluate the performance of an IP-based SCaN network and identify any potential risks. The metrics for performance evaluation are throughput, delay, buffer size, bandwidth constraints and effects of prioritization of data (quality-of-service).

2. SCaN Network Architecture

The SCaN Network architecture [1] for Orion mission will include the Space Network (SN), the Ground Network (GN), and NASA Integrated Services Network.

The space segment of the SN element consists of multiple operational Tracking Data Relay Satellites (TDRSS) in geosynchronous orbit at allocated longitudes for relaying forward and return service signals to and from customers for data transfer and tracking.

The GN sites primarily support S-band communication links. Some GN stations can provide radiometric range and Doppler measurements of space vehicles from the S-band RF links. Some GN ground stations also provide antenna angle tracking data when S-band autotrack is used. The White Sands Complex (WSC) provides the communications equipment necessary for transmitting and receiving data and tracking information relayed via each TDRS. WSC includes three ground terminals which are: the White Sands Ground Terminal (WSGT), the Second TDRSS Ground Terminal (STGT), and the Guam Remote Ground Terminal (GRGT). WSC controls the GRGT remotely because of its location.

The NASA Integrated Services Network (NISN) provides wide area network (WAN) telecommunications services for the transmission of terrestrial data, voice, and video between all SCaN Network ground elements and Constellation/user ground elements.

3. Orion Mission Scenario

During launch and ascent, communication between Orion and Mission Systems is supported by SN (TDRSS) [2]. Mission Systems provides the control center processing and its interfaces with the flight systems for flight operations, crew and flight controller training facilities, mission planning and flight design tools, and personnel for planning, training, flight ops, and mission operations facilities development and maintenance [3]. Communications and Navigation during insertion into orbit and when Orion is in LEO through the completion of docking with ISS will be provided by SCaN through SN (TDRSS). When docked, ISS communications channels are being used where ISS will communicate through TDRSS to relay data to/from Earth. In preparation for Orion separation from ISS, low rate S-band forward and return data services between MS and Orion is reinitiated via the SN. Connectivity is maintained via NISN between GS and MS. During separation, SCaN maintains S-band communication between Orion and MS. During nominal entry, descent, and landing the SCaN network will remain configured for communications and tracking support to Orion until the vehicle reaches the earth's surface and the SCAN is released from Mission support by Constellation.

4. Simulation Tool Environment

The basic software architecture for MACHETE consists of four general systems: (1) orbital and planetary motion kinematics modeling, (2) link engineering modeling, (3) traffic load generation and protocol state machine modeling and execution, and (4) user interface systems spanning all of these three core elements. The resulting combination provides an essential tool for quantifying system performance based on comprehensive considerations of different aspects of space missions. Using this tool, technology researchers and mission designers can (1) determine system resource requirements such as bandwidth, buffer size, and schedule allocations, (2) characterize performance benefits of new or alternative protocols, services, and operations, (3) validate new technologies for mission infusion, and (4) enhance mission planning and operations.

Orbital and Planetary Motion Kinematics Modeling

To provide accurate time-based trajectory data for Orion in this study, a kernel file was prepared using the shuttle mission STS-114 as a model trajectory. This kernel file is a large set of binary numerical data packaged with an internally-invoked numerical interpolation scheme that is readable with various utilities in the JPL Spice software library. The Spice library is a package of programs available for various operating systems and programming environments (including the Matlab scientific computing platform) that one can use to perform many different commonly-encountered tasks in aerospace research. Among the utilities provided are planetary reference frame transformations, planetary position and rotation data in any desired reference frame, conversions for different ephemeris time systems, and many other useful routines.

Once a trajectory kernel is available, coordinates in Cartesian space as a function of time can be read off at any level of resolution desired. For a hypothetical example, if one needs the "x-y-z" positions and velocities of an object specified with respect to the center of the Earth, in the so-called J2000 inertial reference frame, every 1.42 seconds, over the entire mission duration, then a simple call to a Spice routine within the user's executive program will provide this array of data; no further programming or manipulations of data are necessary.

The internal numerical interpolation scheme that runs automatically when the Spice kernel reader calls on the kernel file is the key to providing accurate trajectory data in such a convenient, user-configurable way.

Link Engineering

Multiple additional modeling tools are incorporated within MACHETE for the purpose of characterizing links. The link characterization models include mapping the received signal strength with the waveform modulation to generate a dynamic bit error rate process. Additional examples are to modify a received UHF signal stream to add the stochastic effects of multipath fading, or to capture the effects of weather on a Ka-band deep space link, or to process field data to represent a modified scenario. The data generated from these link characterization tools are fed to the QualNet simulator to incorporate them in simulations of the overall networking behavior.

Traffic and Protocol Modeling

We have developed QualNet models for the complete CCSDS protocol stack, including CCSDS standards Proximity-1[4], Packet Telemetry [5], AOS [6], SCPS [7], and CFDP [8]. The QualNet standard library also contains a full contingent of conventional protocols such as the IEEE 802.11/WiFi and Internet protocol standards.

For traffic generation, QualNet has the capability to model various traffic such as constant bit rate (CBR), and variable bit rate (VBR). The simulation tool can also import traffic profiles generated by other tools, such as by the SCaN traffic studies reported in [9].

5. TRAJECTORIES

Since the precise trajectories for Orion are not available to us to date, we are currently using shuttle mission STS-114 as a sample Orion trajectory in our initial simulation setup. Docking and undocking events are approximated according to previous shuttle-ISS missions; maneuvers during docking and undocking are not currently simulated in detail. Let t_0 denote the instance of start of launch of the ORION. We assume ORION is docked with ISS at 1 day and 20 hours after t_0, and ORION is leaving ISS at 10 days and 16 hours after t_0. Using the trajectory file, we compute the potential communications windows among different systems, as well as the signal propagation delay as spacecraft positions are changing dynamically. The trajectory also impacts the communication paths for network traffic. Tracking and Data Relay Satellites (TDRSS) positions were generated using a standard Keplerian model with an added correction for the lowest-order dynamics induced by the Earth's oblateness; this is the influence from the so-called J2 constant.

The Spice kernel reader is used to replicate the Orion trajectory for the entire mission. However, a corresponding set of accurate time-based position data for the TDRS constellation and for the antennas at the White Sands Complex (WSC) and the GTS site at Guam would still be required as input to network simulator. As this was a nontrivial task for TDRSS, given that the only readily-obtainable TDRSS orbital element data is either for the present time or for the very near-past, we explain the technique used to hypothesize future TDRSS position data in this study.

The mathematical model that was used to obtain TDRSS position data was the standard Keplerian model, based on 7 orbital elements (six constants related to orbit geometry, plus one ephemeris time of the element data “snapshot”), with an added low-order correction for Earth’s oblateness. The orbital elements were taken from NORAD two-line element sets, and were always the quoted mean values only, without the use of the additional perturbation coefficients that NORAD supplies in the sets. The obvious problem in this approach is that the orbital parameters uniquely specify orbiter(s) positions in time, but for TDRSS, only current parameters are available. The mission timeline studied here is several years in the future, so it is necessary to create a hypothetical trajectory set for the TDRSS constellation.

The only reasonable hypothetical TDRSs of the future, given no other information, is that it is identical to the present system. This means that our hypothetical TDRS spacecrafts should be found at positions around the Earth, though moving, that match with those of the present. In fact, when the generation of the trajectories was initiated, “the present” was taken to mean that NORAD orbital parameters were captured within a few hours before running the trajectory program, and those parameters were used to specify the TDRSS constellation’s states for the mission timeline. In this way, all crafts and ground antenna locations in the simulated mission would see TDRSS just as the constellation was the day the trajectory propagator used in this study was run.

Making orbital parameters for a craft “in the present” supply identical Earth-referenced positions at a future time is not merely a case of changing the clock time used in the simulation and running the standard model forward from the desired starting (clock) time. The reason for this is that the Earth rotates independently around its polar axis in the “mean equator” reference frame in which the orbital parameters are specified, so while the satellite trajectories might look nearly the same to the reference frame axes by simply boosting the clock time, the surface of the planet will not have the same relative orientation dynamics with respect to the satellite trajectories. To rectify this situation and thereby effectively create a hypothesized system of TDRSS positions for our analysis of a future mission, it was necessary to counteract the Earth rotation dynamics by a special modification of the TDRSS position data, applied after it was generated by the Keplerian orbit propagator.

The key to understanding how to adjust the propagated TDRSS trajectories so that they replicate correct positions over the Earth's surface in a future time-shifted simulation is to consider that in the mean equator/polar axis reference frame, in which the orbital elements are given, the important motion of the system that is not accounted for is the normal rotation of the planet around the z-axis of the frame. Therefore, a back-rotation must be applied to the satellite position data array to correctly unwind the planet rotation error induced by the time shift, in the equatorial plane, of all of the time-based TDRSS position data. The precise back-rotation angular value was obtained by evaluating data from a JPL Spice kernel file that provides the time-based angular motion models for all of the planets of the Solar System.

For the ground antenna locations involved in our study, those being the White Sands Complex (WSC) and the Guam (GTS) site, the position propagation is only a matter of fixing their locations on the oblate spheroid of the Earth and rotating them around the frame's z-axis according to the Earth's angular rotation model.

6. DYNAMIC NETWORK TOPOLOGY

In the previous section, we described the steps taken to reproduce the trajectories of the elements in the network under study, those being TDRSS, Orion, WSC, and GTS. With all of the trajectories known in the form of positions in Cartesian space at closely spaced instants in time, the next step taken was analyzing all of this motion geometry to determine all of the contact feasibility data. As this is not a trivial task, we next describe the methods used to accomplish this, and the important assumptions employed in that analysis will be highlighted.

A feasible contact between any two communicating assets is deemed in our study to be any time interval during which those two assets have a perfect line-of-sight to each other; this means that both objects are in each others' field of view, and there is no other object (specifically, the Earth) blocking that line-of-sight. Fields of view, for our purposes here, are circular cones of varying vertex angles, although we are immediately capable of considering elliptical and rectangular cones in subsequent studies. Specifically, each ground antenna is assigned a conical field of view that is symmetric around the outward normal vector to the Earth's surface at the antenna location, and the vertex half-angle is always taken to be 80 degrees. This angle provides a 10-degree horizon mask at each ground antenna, such that no orbiting spacecraft could be in the antenna's field of view until it is at least 10 degrees above the geometric horizon plane.

As mentioned previously, Earth is an oblate spheroid (to a good approximation) and we take this into account when computing outward normal vectors for ground antenna contacts and also for occultations to potential contacts between TDRSS assets and Orion. For TDRSS, the field of view for each satellite is set as a cone symmetric about the nadir vector (the vector in the direction of the geometric center of Earth), and TDRSS view cones are set wide-open. That is, each TDRS always has a full 180-degree view, with its antenna boresight pointed straight down at Earth's center. Certainly, Earth itself could block the line-of-sight between TDRS assets and the ORION, but this is the extent of the blockage we assessed in this study. There was not any modeling of signal attenuation through Earth's atmosphere or any other effects besides geometric occultations. For the Orion, we regarded this spacecraft as having an omni-directional antenna, such that it could always feasibly communicate with any other asset in the network that had a line-of-sight contact to it.

In addition to the basic contact feasibility data computed as described above, we computed additional feasibility data for so-called "bent pipe" links of the form Orion-to-TDRS-to-WSC/GTS. These contacts were assessed as "best TDRS" flags to MACHETE, with a "best TDRS" deemed to be, at any time instant, whichever TDRS asset was closest to Orion at that instant and simultaneously in contact with a ground antenna. Both ground antenna sites were independently assessed a best TDRS asset through which to complete an optimal link to Orion at every time instant of the mission scenario. With this information readily available for the entire mission duration, the MACHETE tool always maintained a lookup table of basic network link geometry, as well as geometrically-optimal bent pipe links of shortest propagation delay. This is not to say that the geometrically-optimal bent pipe link was necessarily the one to be used at each instant. In reality, one may need to consider TDRS schedules as well. However, our analysis of the dynamic network geometry included special flagging of the best available bent pipe link at every instant of the mission.

The topology of the network being simulated consists of Orion, Ares, TDRSS, ISS, NISN, mission control centers (MCC) at Johnson Space Center, Marshall, Kennedy Space Center, and Flight Dynamics Facility at Goddard Space Flight Center, and ground stations (as shown in Figure 1). Bandwidths for space links (S-band Low/High data rates) are defined according to concept of operations, SCaN network architecture design and master link book documents. In our initial simulations, we simulated communications between Orion and MCC at JSC using low data rate S-band only.

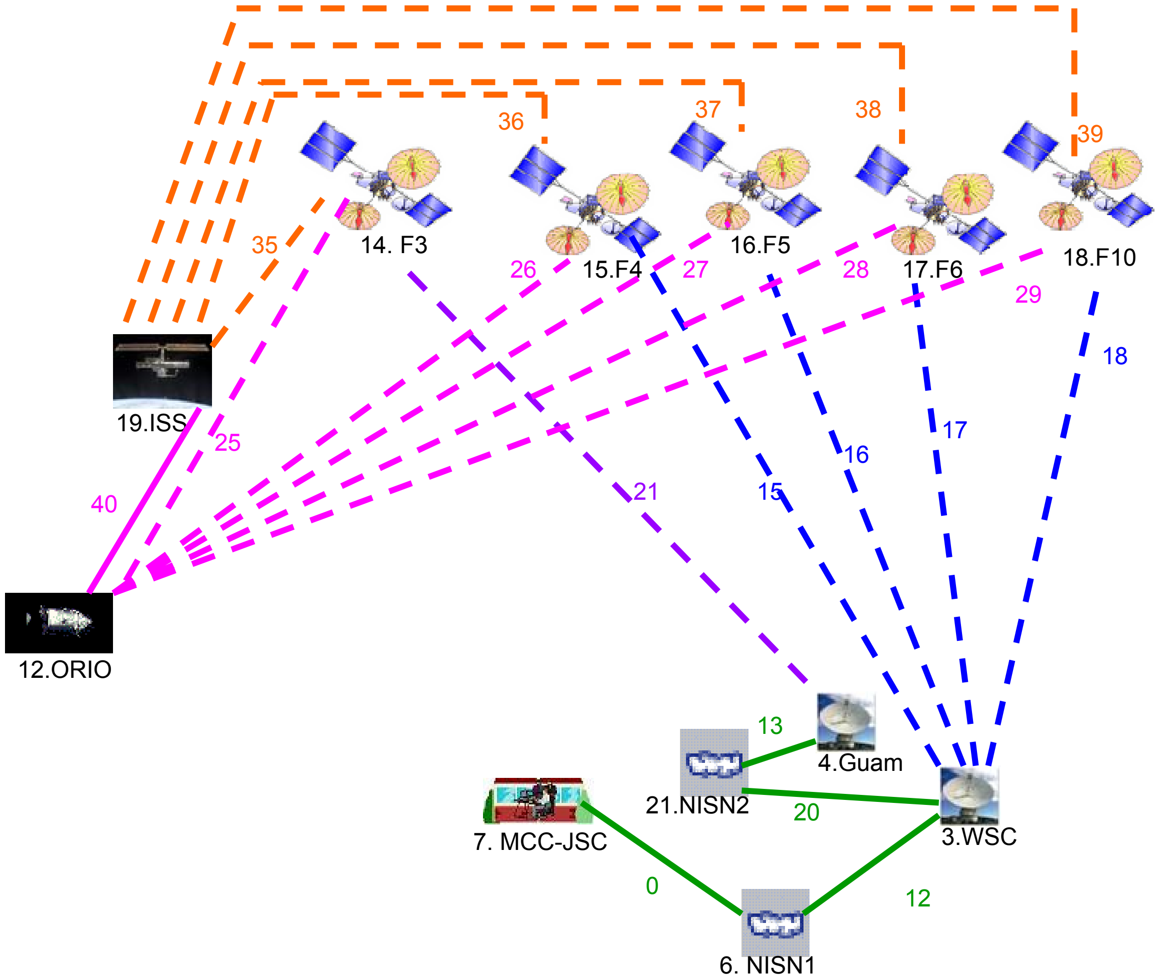


Figure 1 – Orion to ISS Mission Network Topology

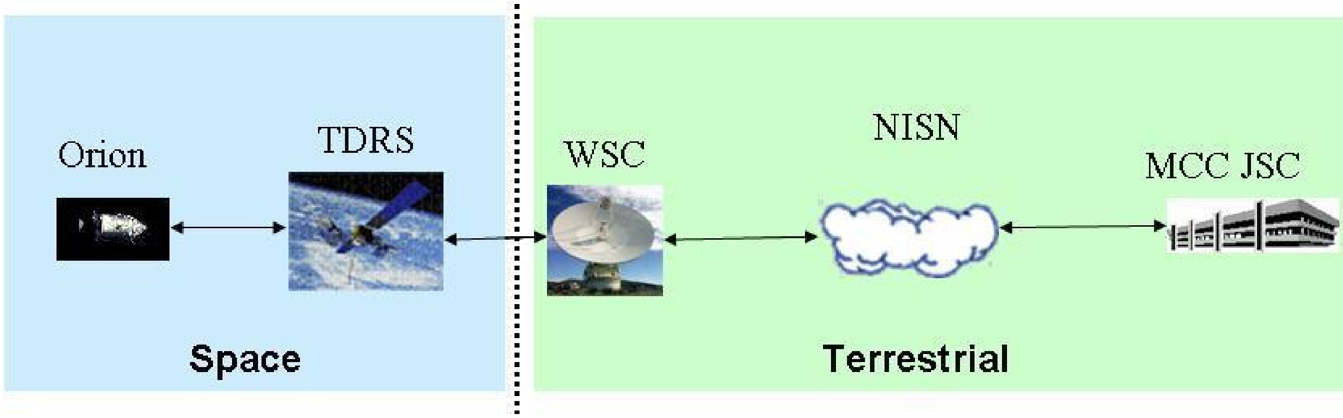
We used the position and visibility analysis to calculate propagation delays with respect of time. We did not model any link errors caused by weather conditions. Forward (MCC to Orion) S-band band-width is 72 kbps; return S-band bandwidth is 192 kbps. We assume a short outage (about 20 seconds) during handover from one TDRS to another. At any instance, we choose the "best TDRS" route from our previously computed table to reach WSC. The route through GTS is used only when there are no available routes through WSC.

7. Communications Protocols

As an initial study of IP-based space networking, we conducted tests to examine data transmission via an architecture using UDP, IP, and AOS. Originally, we planned to simulate the bent pipe through TDRS by using only one IP-address to represent any TDRS. However, if we wish to incorporate the simulation of hand-over from one TDRS to another, we need to use different IP addresses for each TDRS to distinguish them, which is the reason that we are choosing to run IP on TDRS (instead of bent pipe).

Although we are assuming simultaneous end-to-end connectivity, we show that one alternative could be using the Delay Tolerant Bundle Protocol (BP) in case simultaneous end-to-end paths are not available.

As an example, Orion may send telemetry data to Earth using Data Exchange (DE) protocol. This is then transported using UDP and IP. The data units will then flow over the CCSDS AOS protocol between Orion and WSC via TDRS (and sometimes via TDRS and GTS). From WSC, the data are transferred to the appropriate mission control center via NISN.



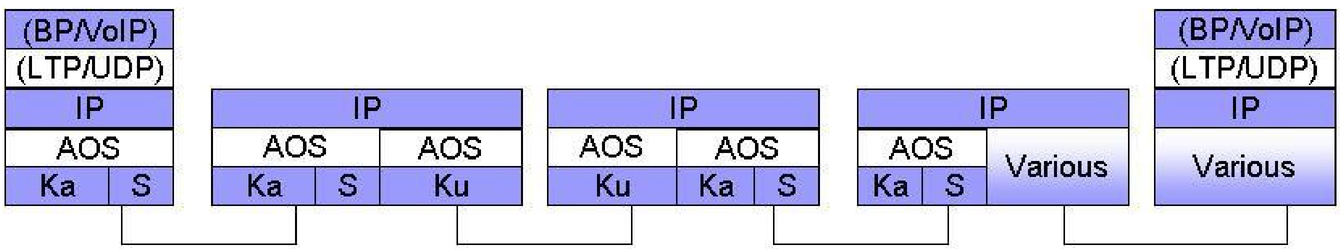


Figure 2 – Example Protocol Stacks

8. NETWORK DATA TRAFFIC MODELS

Four types of traffic are reported in [9], they are voice, real-time motion imagery, telemetry and command. Since we are only considering low data rate S-band links in this simulation, we are only modeling voice, telemetry and command.

In this scenario, the S-band forward link bandwidth (MCC to Orion) is 72 kbps and the return link bandwidth is 192 kbps. During ascent, there is one voice stream, one telemetry stream and one command stream. During Low

Earth Orbit phase, there are two voice streams, one telemetry stream and one command stream. When Orion is docked with ISS, there will be two voice streams. During return to Earth, there will be two voice streams, one telemetry stream and one command stream. Video data is currently not simulated and may be added to the simulation in the future.

Command is modeled as Constant Bit Rate (CBR) traffic at 8 kbps. There are two voice streams. Each voice stream is two way and correlated (meaning each speaker will take turns to talk). The conversations occur at approximately 25% of the time; the silence time is around 75% of the time. We have considered two alternatives for modeling telemetry data. One alternative is to model a single stream of telemetry using CBR at 152.6 kbps. Another alternative is to have two streams of telemetry data: operation telemetry at 58.1 kbps at 50% duty cycle, and engineering telemetry at 152.6 kbps at 50% duty cycle.

9. TWO LOW EARTH ORBIT SCENARIOS

In the first scenario, we are interested in the benefits of using QoS (prioritization of data) to obtain more throughput. The duration for this test is for one day, in Low Earth Orbit. Since our focus is on data prioritization, we are not simulating TDRS handover in this experiment. In this 1-day scenario, the S-band forward link bandwidth is 72 kbps and the return link bandwidth is 192 kbps. Scenario_1 assumes Low Earth Orbit and the Orion is communicating with MCC. The traffic streams are as follows:

(1) One command stream, modeled as CBR at 8 kbps.

(2) Two voice streams, modeled using gamma distribution, where the talk time has a mean of 5.86 seconds and a variance of 16.1 seconds and quiet time has a mean of 7.47 seconds and a variance of 19.13 seconds. The peak is at 19.8 kbps and the conversation length has a distribution mean of 10 minutes.

(3) One telemetry stream, modeled as CBR at 152.6 kpbs.

(4) One delay-tolerant stream, modeled as CBR at 30 kbps.

Two cases were run. In the first case (Case_A), command and voice streams are assigned the highest priority. Telemetry is assigned medium priority and delay-tolerant traffic is assigned low priority. We set the queue sizes for high, medium and low priorities are set at 10,000 bytes, 30,000 bytes and 10,000 bytes respectively.

In the second case (Case_B), the traffics are the same as Case_A; however, all traffic streams are of the same priority and the queue size is 50,000 bytes (which is equal to the sum of the queue sizes in Case_A).

On the return link, adding the peak rates of the two voice streams and telemetry results in 192.2 kbps; which would saturate the return link. However, since there are times when the voice stream is quiet, we could use the bandwidth for other types of traffic. We added another 30 kbps of low priority delay tolerant traffic. That brings the total peak traffic to 222.2 kbps which is 15.7% in excess of the 192 kbps bandwidth.

With prioritization, Case_A shows that we could fit another 27 kbps of low priority traffic without loss to any other traffic. More specifically, 93% of the 30 kbps low priority traffic and 93% went through. The average delay for high priority data is less than 0.41 seconds, where the propagation delay is approximately 0.26 seconds so the queuing delay is at most 0.15 seconds.

Without prioritization, Test_B lost 4% of telemetry and approximately 2% on each of the voice streams on the return link. However, comparing the total bit loss in Case A and Case B, the loss in Case A is approximately 23 megabits while in Case B, it is 59 megabits. Thus, using separate buffers for each priority increased the total throughput.

Scenario_2 is similar to the first scenario except the duration of the test is 1 hour and the traffic streams are not the same as Scenario_1. In this scenario, we modeled two telemetry streams with different priorities. There is one command stream modeled as CBR at 8 kbps. There are two voice streams at peak rate of 19.8 kbps. Command and voice streams are assigned high priority. However, we separated telemetry into two streams: operation telemetry at 58.1 kbps at 50% duty cycle and engineering telemetry at 152.6 kbps at 50% duty cycle. The operation telemetry has medium priority and the engineering telemetry has low priority. There is no delay-tolerant traffic. Adding the peak traffic values on the return link gives 250.3 kbps which exceeds the return link bandwidth of 192 kbps by 30.4%. However, since voice traffic has quiet time, and telemetry streams have a 50% duty cycle, all data of high and medium priorities were delivered. There is only a 5% loss on the low priority telemetry stream.

10. NOMINAL ASCENT TO RETURN SCENARIO

A nominal Orion to ISS mission scenario was simulated as follows. Using the STS-114 trajectory file, together with the predicted TDRSS trajectories, we obtained a contact table with file name "Network_Contacts_114.txt" that lists the contact range information for links between ORION and various TDRS (F3—F6, F10) satellites, as well as links between TDRS and the two ground sites, WSC and Guam. The range values are in kilometers; a range value of 0 means the link is occulted. Range and TDRS information are updated every minute. The contact information is used in configuring the link on/offs in the simulation configuration file and in determining communication routes. Previous mission data is used to estimate docking and undocking time; thus controlling the link (on/off) between Orion and ISS.

We used the position and visibility analysis to calculate propagation delays with respect to time. We did not model any link errors caused by weather conditions. Forward (MCC to Orion) S-band band-width is assumed to be 72 kbps; return S-band bandwidth is assumed to be 192 kbps. We assume switching among TDRSS to be instantaneous. At any instance, we choose the "best TDRS" route from our previously computed table to reach WSC. The route through GTS is used only when there are no available routes through WSC. From the range information, we compute propagation delays between Orion and each TDRS (F3 – F6, F10) with respect to time, updated at every minute. The distance between TDRS and Guam (GTS), or TDRS and WSC do not change significantly (at approximately 40,000 km); the one way propagation delay is assumed to be approximately 0.13 second.

To model a bent-pipe, one approach is to use one simulated node to represent all the TDRS satellites. However, if we intend to simulate switching among different TDRS satellites, the network simulation tool poses certain limitations. Currently, we do not have a link layer model for switching among the satellites. Thus, we are using network layer routing to model TDRS switch over. To do this, we need to provide a routing table with respect to each dynamic topology change. There are two sets of five routing tables; one set for Orion when it is docked with ISS, the other set for Orion when it is not docked with ISS. Each set of contains five routing tables, one for each possible TDRS (F3—F6, F10). The routing table size is O(VE), where V is the number of nodes and E the number of edges/links. In Figure 1, there are 12 nodes and 20 edges, so each routing table contains 240 entries. In a 15-day mission scenario, there are approximately 900 topology changes due to switching among TDRS satellites. Based on the topology, we generate dynamic routes with respect to time in the simulation configuration file. More specifically, we compute routing tables with respect to time (scheduled routes).

The communications protocols used in this experiment are:

- Application Layer
    - Constant Bit Rate (CBR): to model telemetry and command.
    - Traffic Generation Application: to model voice traffic
- Transport Layer: User Datagram Protocol (UDP)
- Network Layer:
    - Internet Protocol (IP)
    - Routing protocol is scheduled route. That is, we update the routing table according to scheduled route with respect to time.
- Link Layer:
    - Wired Link: for terrestrial connections
    - Space Link: custom model in MACHETE
- Physical Layer: currently not modeled in detail. Some physical effects are modeled in the Space Link model.

Three types of traffic are simulated. In the forward direction (from MCC to Orion), there is one stream of command. In the return direction (from Orion to MCC), there is one stream of telemetry. In addition to these, there are two voice streams, where each stream is 2-way (for conversation). Telemetry is modeled as CBR at peak rate of 152.6 kbps from Orion to MCC. Command is modeled as CBR at peak rate of 8 kbps from MCC to Orion. The two voice streams are assumed to have peak rate of 10 kbps assuming there is on-going traffic all the time. The talk time has a mean of 5.86 seconds and quiet time has a mean of 7.47 seconds (according to [9]). Probability of data generation is 0.45. The expected volume of voice traffic in each voice stream for each direction is estimated to be (10 kbps * (5.86) / (5.86+7.47)) = 1.95 kbps. There are a couple of limitations in the current simulation tool: (a) the 2-way voice traffic is not correlated, (b) the traffic is generated in large bursts (58600 bits) with random exponential distribution. The interval between bursts is generated with exponential distribution, where the mean interval time is 13.33 seconds. Fragment size for traffic generation application is set to be at 65022 Bytes. In a 15-day scenario, the amount of data generated is large. Approximately 14 Gigabits of command is sent and 189 Gigabits of telemetry is sent.

Simulation result showed the following. For telemetry, there is a 12% data loss; throughput is at 134.6 kbps; average end-to-end delay is 1.24 second; average jitter is 0.3 second.

For command, there is a 6% data loss; throughput is at 7.5 kbps; average end-to-end delay is 0.64 second; average jitter is 0.26 second.

There are two voice streams. Both streams are generated using the same parameters thus the statistics are also similar. For both voice streams, in the forward direction, there is a 6% data loss; throughput is at 1.9 kbps; average end-to-end delay is 1.49 second; average jitter is 1.07 second. In the return direction, there is a 12% data loss; throughput is at 1.76 kbps; average end-to-end delay is 1.07 second; average jitter is 1.07 second.

Looking at more detailed statistics collected at each node, we observed that most of the data loss is due to "IP no-route". The data drops occur at the TDRS satellites and at the ISS. This is due to a TDRS still has packets to forward, but the scheduled route is switched. Thus the remaining packets cached cannot be forwarded. Another small amount of data drops is due to fragments being dropped (about 0.017% of total drops).

11. CONCLUSION

In this paper, we described our initial work on simulating the SCaN network for the Orion to ISS mission. The process involved obtaining realistic mission trajectories, calculating the dynamic network topologies, characterizing traffic patterns, determining protocols to be used in the scenarios modeled. In the LEO scenarios, we held the network topology static and compared the cases where data are assigned different priorities versus treating all data as having same priority. We observed that, when priorities are assigned, we suffer less data loss and more efficient use of the available bandwidth. In the nominal Orion to ISS 15-day mission scenario, we looked at using IP and scheduled route for communication where the telemetry, command and voice data are all at the same priority. We observed that there is considerable data loss due to switching among different TDRS satellites instantaneously (without overlap handover).

In future work, we would like to extend the work in the following aspects. It would be desirable to obtain more accurate information on Orion trajectory (instead of using previous shuttle trajectories). We are currently building a more detailed traffic generation application for voice and video according to the result in [9]. This model will be incorporated into the simulation tool as add-on models. We are also currently developing link budget libraries. Another extension to the tool is a better TDRS model and more accurate modeling of TDRS hand-over.

ACKNOWLEDGEMENTS

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology and the National Aeronautics and Space Administration.

REFERENCES

[1] K. Bhasin, J. Hudiburg, R. Miller, "Space Communications and Navigation Constellation Integration Project -- Architecture Design Document: ORION-ISS Mission Phase" (Draft), March 2007.

[2] B. Castle, J. Hanley, P. Vrotsos, "Joint Constellation (Cx) – Space Communications and Navigation (SCaN) Concept of Operations" (Draft), October, 2007.

[3] "Constellation Architectures Requirement Document," (CxP 70000), December 2006.

[4] CCSDS 211.0-B-3. Proximity-1 Space Link Protocol—Data Link Layer. Blue Book. Issue 3. May 2004.

[5] CCSDS 132.0-B-1. TM Space Data Link Protocol. Blue Book. Issue 1. September 2003.

[6] CCSDS 714.0-B-1. Space Communications Protocol Specification (SCPS)—Transport Protocol (SCPS-TP). Blue Book. Issue 1. May 1999.

[7] CCSDS 732.0-B-1. AOS Space Data Link Protocol. Blue Book. Issue 1. September 2003.

[8] CCSDS 727.0-B-3. CCSDS File Delivery Protocol (CFDP). Blue Book. Issue 3. June 2005.

[9] T. Stoenescu, L. Clare, "Traffic Modeling for NASA's Space Communications and Navigation (SCaN) Network", submitted to 2008 IEEE Aerospace Conference Proceedings, March 1—8, 2008.

BIOGRAPHY

Esther Jennings *is a research staff with the Communication Systems and Research Section of the Jet Propulsion Laboratory. She received a Ph.D. in Computer Science from Luleå University of Technology, Sweden in 1997. From 1997-1999, she was a postdoctoral fellow at the Industrial Engineering Department at Technion, Israel Institute of Technology. From 1999-2001, she has been an assistant professor at the Computer Science Department of California State Polytechnic University, Pomona. Her research interests are in distributed graph algorithms, reliable multicast protocols, energy-efficient algorithms for wireless network and algorithm simulations.*

David Heckman *is a mathematician at the Telecommunications Architecture Group, at the Jet Propulsion Laboratory. He is currently pursuing a Ph.D. in Mathematics at Claremont Graduate University. His thesis studies are in the areas of stochastic optimization algorithms and optimal scheduling/sequencing problems, with emphasis on Markov Chain Monte Carlo algorithms. His other research interests include general optimization, numerical optimization algorithms, Bayesian inference, statistics, general numerical analysis, the calculus of variations, ballistics, probability, and stochastic processes. At JPL, he is actively involved in the development of the proprietary TOAST telecom/orbital analysis tool and regularly develops customized simulation engines for analyzing various telecommunications architectures and mission scenario*